On the guaranteed error correction capability of LDPC codes

Shashi Kiran Chilappagari, Dung Viet Nguyen, Bane Vasić, Michael Marcellin
Dept. of Electrical and Computer Eng.
University of Arizona
Tucson, AZ 85721, USA
Email: {shashic,nguyendv,vasic,marcellin}@ece.arizona.edu

Abstract— We investigate the relation between the girth and the guaranteed error correction capability of γ-left regular LDPC codes when decoded using the bit flipping (serial and parallel) algorithms. A lower bound on the number of variable nodes which expand by a factor of at least $3\gamma/4$ is found based on the Moore bound. An upper bound on the guaranteed correction capability is established by studying the sizes of smallest possible trapping sets.

I. Introduction

Iterative algorithms for decoding low-density parity-check (LDPC) codes have been the focus of research over the past decade and most of their properties are well understood [1], [2]. These algorithms operate by passing messages along the edges of a graphical representation of the code known as the Tanner graph, and are optimal when the underlying graph is a tree. Message passing decoders perform remarkably well which can be attributed to their ability to correct errors beyond the traditional bounded distance decoding capability. However, in contrast to bounded distance decoders (BDDs), the guaranteed error correction capability of iterative decoders is largely unknown.

The problem of recovering from a fixed number of erasures is solved for iterative decoding on the binary erasure channel (BEC). If the Tanner graph of a code does not contain any stopping sets [3] up to size t (or equivalently the size of minimum stopping set is $t+1$), then the decoder is guaranteed to recover from any t erasures. Orlitsky *et al.* in [4] studied the relation between stopping sets and girth and derived bounds on the smallest stopping set in any d-left regular bipartite graph and girth g.

An analogous result is unknown for decoding on other channels such as the binary symmetric channel (BSC) and the additive white Gaussian noise (AWGN) channel. In this paper, we present a step toward such result for hard decision decoding algorithms. Gallager [5] proposed two binary message passing algorithms, namely Gallager A and Gallager B, for decoding over the BSC. He showed that for the column-weight $\gamma \geq 3$ and $\rho > \gamma$, there exist (n, γ, ρ) [1] regular low-density parity-check (LDPC) codes for which the bit error probability asymptotically tends to zero whenever we operate below the threshold. The minimum distance was shown to

increase linearly with the code length, but correction of a linear fraction of errors was not shown. Zyablov and Pinsker [6] analyzed LDPC codes under a simpler decoding algorithm known as the bit flipping algorithm and showed that almost all the codes in the regular ensemble with $\gamma \geq 5$ can correct a constant fraction of worst case errors. Sipser and Spielman in [7] used expander graph arguments to analyze two bit flipping algorithms, serial and parallel. Specifically, they showed that these algorithms can correct a fraction of errors if the underlying Tanner graph is a good expander. Burshtein and Miller in [8] applied expander based arguments to show that message passing algorithms can also correct a fixed fraction of worst case errors when the degree of each variable node is more than five. Feldman *et al.* [9] showed that the linear programming decoder [10] is also capable of correcting a fraction of errors. Recently, Burshtein in [11] showed that regular codes with variable nodes of degree four are capable of correcting a linear number of errors under bit flipping algorithm. He also showed tremendous improvement in the fraction of correctable errors when the variable node degree is at least five.

It is well known that a random graph is a good expander with high probability [7]. However, the fraction of nodes having the required expansion is very small and hence the code length to guarantee correction of a fixed number of errors must be large. Moreover, determining the expansion of a given graph is known to be NP hard [12], and spectral gap methods cannot guarantee an expansion factor of more than 1/2 [7]. On the other hand, code parameters such as column weight and girth can be easily determined or are assumed to be known for the code under consideration. The approach in this paper is to determine the size of variable node sets in a left regular LDPC code which are guaranteed to have the expansion required by bit flipping algorithms based on Moore bound [13, p.180]. The consequence of our results is that the error correction capability grows exponentially in girth. However, we note that since the girth grows logarithmically in the code length, this result does not show that the bit flipping algorithms can correct a linear fraction of errors (the proof and discussion are beyond the scope of this paper).

To find an upper bound on the number of correctable errors, we study the size of sets of variable nodes which lead to decoding failures. A decoding failure is said to have occurred

---

[1]Precise definitions will be given in Section II

if the output of the decoder is not equal to the transmitted codeword [14]. The conditions that lead to decoding failures are well understood for a variety of decoding algorithms such as maximum likelihood decoding, bounded distance decoding and iterative decoding on the binary erasure channel. However, for iterative decoding on the BSC and AWGN channel, the understanding is far from complete. Two approaches have been taken in this direction, namely trapping sets [14] and pseudo-codewords [15]. We adopt the trapping set approach in this paper to characterize decoding failures. Richardson introduced the notion of trapping sets in [14] to estimate the error floor on the AWGN channel. In [16], trapping sets were used to estimate the frame error rate of column-weigh-three LDPC codes. In this paper, we define trapping sets with the help of fixed points for the bit flipping algorithms (both serial and parallel). We then find bounds on the size of trapping sets based on extremal graphs known as cage graphs [17], thereby finding an upper bound on the guaranteed error correction capability.

The rest of the paper is organized as follows. In Section II, we provide a brief introduction to LDPC codes, decoding algorithms and trapping sets [14]. In Section III, we prove our main theorem relating the column weight and girth to the number of variable nodes which expand by a factor of at least $3\gamma/4$. We derive bounds on the size of trapping sets in Section IV, and conclude with a few remarks in Section V.

II. PRELIMINARIES

In this section, we first establish the notation and then proceed to give a brief introduction to LDPC codes and bit flipping algorithms. We then give the relation between the error correction capability of the code and the expansion of the underlying Tanner graph. We finally describe trapping sets for the bit flipping algorithms.

A. Graph Theory Notation

We adopt the standard notation in graph theory (see [18] for example). $G = (U, E)$ denotes a graph with set of nodes U and set of edges E. When there is no ambiguity, we simply denote the graph by G. An edge e is an unordered pair (u_1, u_2) of nodes and is said to be incident on u_1 and u_2. Two nodes u_1 and u_2 are said to be adjacent (neighbors) if there is an edge $e = (u_1, u_2)$ incident on them. The order of the graph is $|U|$ and the size of the graph is $|E|$. The degree of u, $d(u)$, is the number of its neighbors. A node with degree one is called a leaf or a pendant node. A graph is d-regular if all the nodes have degree d. The average degree $\overline{d}$ of a graph is defined as $\overline{d} = 2|E|/|U|$. The girth $g(G)$ of a graph G, is the length of smallest cycle in G. $H = (V \cup C, E')$ denotes a bipartite graph with two sets of nodes; variable (left) nodes V and check (right) nodes C and edge set E'. Nodes in V have neighbors only in C and vice versa. A bipartite graph is said to be γ-left regular if all variable nodes have degree γ, ρ-right regular if all check nodes have degree ρ and (γ, ρ) regular if all variable nodes have degree γ and all check nodes have degree ρ. The girth of a bipartite graph is even.

B. LDPC Codes and Decoding Algorithms

LDPC codes [5] are a class of linear block codes which can be defined by sparse bipartite graphs [19]. Let G be a bipartite graph with two sets of nodes: n variable nodes and m check nodes. This graph defines a linear block code $\mathcal{C}$ of length n and dimension at least $n - m$ in the following way: The n variable nodes are associated to the n coordinates of codewords. A vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is a codeword if and only if for each check node, the modulo two sum of its neighbors is zero. Such a graphical representation of an LDPC code is called the Tanner graph [20] of the code. The adjacency matrix of G gives a parity check matrix of $\mathcal{C}$. An (n, γ, ρ) regular LDPC code has a Tanner graph with n variable nodes each of degree γ (column weight) and $n\gamma/\rho$ check nodes each of degree ρ (row weight). This code has length n and rate $r \geq 1 - \gamma/\rho$ [19].

We now describe the parallel bit flipping algorithm [6], [7] to decode LDPC codes. As noted earlier, each check node imposes a constraint on the neighboring variable nodes. A constraint (check node) is said to be satisfied by a setting of variable nodes if the sum of the variable nodes in the constraint is even; otherwise the constraint is unsatisfied.

Parallel Bit Flipping Algorithm
- In parallel, flip each variable that is in more unsatisfied than satisfied constraints.
- Repeat until no such variable remains.

A serial version of the algorithm is also defined in [7] and all the results in this paper hold for the serial bit flipping algorithm also. The bit flipping algorithms are iterative in nature but do not belong to the class of message passing algorithms.

C. Expansion and Error Correction Capability

Sipser and Spielman [7] analyzed the performance of the bit flipping algorithms using the expansion properties of the underlying Tanner graph of the code. We summarize the results from [7] below for the sake of completeness. We start with the following definitions from [7].

Definition 1: Let $G = (U, E)$ with $|U| = n_1$. We say that *every set of at most m_1 nodes expands by a factor of δ* if, for all sets $S \subset U$

$$|S| \leq m_1 \Rightarrow |\{y : \exists x \in S \text{ such that } (x, y) \in E\}| > \delta|S|.$$

We consider bipartite graphs and expansion of variable nodes only.

Definition 2: A graph is a $(\gamma, \rho, \alpha, \delta)$ expander if it is a (γ, ρ) regular bipartite graph in which every subset of at most α fraction of the variable nodes expands by a factor of at least δ.

The following theorem from [7] relates the expansion and error correction capability of an (n, γ, ρ) LDPC code with Tanner graph G when decoded using the parallel bit flipping decoding algorithm.

Theorem 1: [7, Theorem 11] Let G be a $(\gamma, \rho, \alpha, (3/4 + \epsilon)\gamma)$ expander over n variable nodes, for any $\epsilon > 0$. Then, the simple parallel decoding algorithm will correct any $\alpha_0 <$

$\alpha(1 + 4\epsilon)/2$ fraction of errors after $\log_{1-4\epsilon}(\alpha_0 n)$ decoding rounds.

Notes:

1) The serial bit flipping algorithm can also correct $\alpha_0 < \alpha/2$ fraction of errors if G is a $(\gamma, \rho, \alpha, (3/4)\gamma)$ expander.
2) The results hold for any left regular code as we need expansion of variable nodes only.

From the above discussion, we observe that finding the number of variable nodes which are guaranteed to expand by a factor of at least $3\gamma/4$, gives a lower bound on the guaranteed error correction capability of LDPC codes.

D. Decoding Failures and Trapping Sets

We now define decoding failures of the bit flipping algorithms and characterize these failures using trapping sets.

Consider an LDPC code of length n and let $\mathbf{x}$ be the binary vector which is the input to the hard decision decoder. Let $\mathcal{S}(\mathbf{x})$ be the support of $\mathbf{x}$. The support of $\mathbf{x}$ is defined as the set of all positions i where $\mathbf{x}(i) \neq 0$. The set of variable nodes (bits) which differ from their original value are referred to as corrupt variables.

Definition 3: $\mathbf{x}$ is a fixed point of the bit flipping algorithm if the set of corrupt variables remains unchanged after one round of decoding.

Definition 4: [16] Let $\mathbf{x}$ be a fixed point. Then $\mathcal{S}(\mathbf{x})$ is known as a trapping set. An (a, b) trapping set $\mathcal{T}$ is a set of a variable nodes whose induced subgraph has b odd degree checks.

The size of $\mathcal{T}$ denoted by $|\mathcal{T}|$ is the number of variable nodes in $\mathcal{T}$. From the definitions, it is clear that if the input to the decoder is a fixed point then the output is also the same fixed point. Or in other words, if the initial errors are in variable nodes corresponding to a trapping set, then the decoder will not converge to the original codeword. Hence, bounding the size of trapping sets gives an upper bound on the guaranteed error correction capability. Apart from fixed point decoding failures, there exist other types of failures, the discussion of which is beyond the scope of this paper.

III. EXPANSION, COLUMN WEIGHT AND GIRTH

In this section, we prove our main theorem which relates the column weight and girth of a code to its error correction capability. We show that the size of variable node sets which have the required expansion is related to the well known Moore bound [13, p.180]. We start with a few definitions required to establish the main theorem.

A. Definitions

Definition 5: The *reduced graph* $H_r = (V \cup C_r, E'_r)$ of $H = (V \cup C, E')$ is a graph with vertex set $V \cup C_r$ and edge set E'_r given by

$$C_r = C \setminus C_p, \ C_p = \{c \in C : \text{c is a pendant node}\}$$
$$E'_r = E' \setminus E'_p, \ E'_p = \{(v_i, c_j) \in E : c_j \in C_p\}$$

Definition 6: Let $H = (V \cup C, E')$ be such that $\forall v \in V, d(v) \leq \gamma$. The γ *augmented graph* $H_\gamma = (V \cup C_\gamma, E'_\gamma)$ is a graph with vertex set $V \cup C_\gamma$ and edge set E'_γ given by

$$C_\gamma = C \cup C_a, \text{ where } C_a = \bigcup_{i=1}^{|V|} C_a^i \text{ and}$$
$$C_a^i = \{c_1^i, \ldots, c_{\gamma-d(v_i)}^i\}.$$
$$E'_\gamma = E' \cup E'_a, \text{ where } E'_a = \bigcup_{i=1}^{|V|} E_a^{'i} \text{ and}$$
$$E_a^{'i} = \{(v_i, c_j) \in V \times C_a : c_j \in C_a^i\}.$$

Definition 7: [7, Definition 4] The *edge-vertex incidence graph* $G_{ev} = (U \cup E, E_{ev})$ of $G = (U, E)$ is the bipartite graph with vertex set $U \cup E$ and edge set

$$E_{ev} = \{(e, u) \in E \times U : u \text{ is an endpoint of e}\}.$$

Notes:

1) The edge-vertex incidence graph is right regular with degree two.
2) $|E_{ev}| = 2|E|$.
3) $g(G_{ev}) = 2g(G)$.

Definition 8: An *inverse edge-vertex incidence graph* $H_{iev} = (V, E'_{iev})$ of $H = (V \cup C, E')$ is a graph with vertex set V and edge set E'_{iev} which is obtained as follows. For $c \in C_r$, let $N(c)$ denote the set of neighbors of c. Label one node $v_i \in N(c)$ as a root node. Then

$$E'_{iev} = \{(v_i, v_j) \in V \times V : v_i \in N(c), v_j \in N(c),$$
$$i \neq j, \ v_i \text{ is a root node, for some } c \in C_r\}.$$

Notes:

1) Given a graph, the inverse edge-vertex incidence graph is not unique.
2) $g(H_{iev}) \geq g(H)/2, |E'_{iev}| = |E'_r| - |C_r|$ and $|C_r| \leq |E'_r|/2$
3) $|E'_{iev}| \geq |E'_r|/2$ with equality only if all checks in C_r have degree two.
4) The term inverse edge-vertex incidence is used for the following reason. Suppose all checks in H have degree two. Then the edge-vertex incidence graph of H_{iev} is H.

The *Moore bound* [13, p.180] denoted by $n_0(d, g)$ is a lower bound on the least number of vertices in a d-regular graph with girth g. It is given by

$$n_0(d, g) = n_0(d, 2r+1) = 1 + d\sum_{i=0}^{r-1}(d-1)^i, g \text{ odd}$$

$$n_0(d, g) = n_0(d, 2r) = 2\sum_{i=0}^{r-1}(d-1)^i, g \text{ even}$$

In [21], it was shown that a similar bound holds for irregular graphs.

Theorem 2: [21] The number of nodes $n(\overline{d}, g)$ in a graph of girth g and average degree at least $\overline{d} \geq 2$ satisfies:

$$n(\overline{d}, g) \geq n_0(\overline{d}, g)$$

Note that $\overline{d}$ need not be an integer in the above theorem.

B. The Main Theorem

We now state and prove the main theorem.

Theorem 3: Let $\mathcal{C}$ be an LDPC code with γ-left regular Tanner graph G. Let $g(G) = 2g'$. Then for all $k < n_0(\gamma/2, g')$, any set of k variable nodes expands by a factor of at least $3\gamma/4$.

Proof: Let $G^k = (V^k \cup C^k, E^k)$ denote the subgraph induced by a set of k variable nodes V^k. Since G is γ-left regular, $|E^k| = \gamma k$. Let $G_r^k = (V^k \cup C_r^k, E_r^k)$ be the reduced graph. We have

$$
\begin{aligned}
|C^k| &= |C_r^k| + |C_p^k| \\
|E^k| &= |E_p^k| + |E_r^k| \\
|E_p^k| &= |C_p^k| \\
|C_p^k| &= \gamma k - |E_r^k|
\end{aligned}
$$

We need to prove that $|C^k| > 3\gamma k/4$.

Let $f(k, g')$ denote the maximum number of edges in an arbitrary graph of order k and girth g'. By Theorem 2, for all $k < n_0(\gamma/2, g')$, the average degree of a graph with k nodes and girth g' is less than $\gamma/2$. Hence, $f(k, g') < \gamma k/4$. We now have the following lemma.

Lemma 1: The number of edges in G_r^k cannot exceed $2f(k, g')$ i.e.,

$$|E_r^k| \leq 2f(k, g').$$

Proof: The proof is by contradiction. Assume that $|E_r^k| > 2f(k, g')$. Consider $G_{iev}^k = (V^k, E_{iev}^k)$, an inverse edge vertex incidence graph of G^k. We have

$$|E_{iev}^k| > f(k, g').$$

This is a contradiction as G_{eiv}^k is a graph of order k and girth at least g'. ∎

We now find a lower bound on $|C^k|$ in terms of $f(k, g')$. We have the following lemma.

Lemma 2: $|C^k| \geq \gamma k - f(k, g')$.

Proof: Let $|E_r^k| = 2f(k, g') - j$ for some integer $j \geq 0$. Then $|E_p^k| = \gamma k - 2f(k, g') + j$. We claim that $|C_r^k| \geq f(k, g') + j$. To see this, we note that

$$
\begin{aligned}
|E_{iev}^k| &= |E_r^k| - |C_r^k|, \text{ or} \\
|C_r^k| &= |E_r^k| - |E_{iev}^k|
\end{aligned}
$$

But

$$
\begin{aligned}
|E_{iev}^k| &\leq f(k, g') \\
\Rightarrow |C_r^k| &\geq 2f(k, g') - j - f(k, g') \\
\Rightarrow |C_r^k| &\geq f(k, g') - j
\end{aligned}
$$

Hence we have,

$$
\begin{aligned}
|C^k| &= |C_r^k| + |C_p^k| \\
\Rightarrow |C^k| &\geq f(k, g') - j + \gamma k - 2f(k, g') + j \\
\Rightarrow |C^k| &\geq \gamma k - f(k, g').
\end{aligned}
$$

∎

The theorem now follows as

$$f(k, g') < \gamma k/4$$

and therefore

$$|C^k| > 3\gamma k/4$$

∎

Corollary 1: Let $\mathcal{C}$ be an LDPC code with column-weight γ and girth $2g'$. Then the bit flipping algorithm can correct any error pattern of weight less than $n_0(\gamma/2, g')/2$.

IV. CAGE GRAPHS AND TRAPPING SETS

In this section, we first give necessary and sufficient conditions for a given set of variables to be a trapping set. We then proceed to define a class of interesting graphs known as cage graphs and establish a relation between cage graphs[17] and trapping sets. We then give an upper bound on the error correction capability based on the sizes of cage graphs. The proofs in this section are along the same lines as in Section III. Hence, due to space considerations, we only give a sketch of the proofs.

Theorem 4: Let $\mathcal{C}$ be an LDPC code with γ-left regular Tanner graph G. Let $\mathcal{T}$ be a set consisting of V variable nodes with induced subgraph $\mathcal{I}$. Let the checks in $\mathcal{I}$ be partitioned into two disjoint subsets; $\mathcal{O}$ consisting of checks with odd degree and $\mathcal{E}$ consisting of checks with even degree. Then $\mathcal{T}$ is a trapping set for bit flipping algorithm iff : (a) Every variable node in $\mathcal{I}$ has at least $\lceil \gamma/2 \rceil$ neighbors in $\mathcal{E}$, and (b) No $\lfloor \gamma/2 \rfloor + 1$ checks of $\mathcal{O}$ share a neighbor outside $\mathcal{I}$.

Proof: We first show that the conditions stated are sufficient. Let $\mathbf{x}_\mathcal{T}$ be the input to the bit flipping algorithm, with support $\mathcal{T}$. The only unsatisfied constraints are in $\mathcal{O}$. By the conditions of the theorem, we observe that no variable node is involved in more unsatisfied constraints than satisfied constraints. Hence, no variable node is flipped and by definition $\mathbf{x}_\mathcal{T}$ is a fixed point implying that $\mathcal{T}$ is a trapping set.

To see that the conditions are necessary, observe that for $\mathbf{x}_\mathcal{T}$ to be a trapping set, no variable node should be involved in more unsatisfied constraints than satisfied constraints. ∎

Remark: Theorem 4 is a consequence of Fact 3 from [14].

To determine whether a given set of variables is a trapping set, it is necessary to not only know the induced subgraph but also the neighbors of the odd degree checks. However, in order to establish general bounds on the sizes of trapping sets given only the column weight and the girth, we consider only condition (a) of Theorem 4 which is a necessary condition. A set of variable nodes satisfying condition (a) is known as a *potential trapping set*. A trapping set is a potential trapping set that satisfies condition (b). Hence, finding bounds on the size of potential trapping sets gives bounds on the size of trapping sets.

Definition 9: [17] A (d, g)-*cage graph*, $G(d, g)$, is a d-regular graph with girth g having the minimum possible number of nodes.

A lower bound, $n_l(d, g)$, on the number of nodes $n_c(d, g)$ in a (d, g)-cage graph is given by the Moore bound. An upper bound $n_u(d, g)$ on $n_c(d, g)$ (see [17] and references therein)

is given by

$$n_u(3,g) = \begin{cases} \frac{4}{3} + \frac{29}{12} \, 2^{g-2} & \text{for g odd} \\ \frac{2}{3} + \frac{29}{12} \, 2^{g-2} & \text{for g even} \end{cases}$$

$$n_u(d,g) = \begin{cases} 2(d-1)^{g-2} & \text{for g odd} \\ 4(d-1)^{g-3} & \text{for g even} \end{cases}$$

Theorem 5: Let $\mathcal{C}$ be an LDPC code with γ-left regular Tanner graph G and girth $2g'$. Let $\mathcal{T}(\gamma, 2g')$ denote the the smallest possible trapping set of $\mathcal{C}$ for the bit flipping algorithm. Then,

$$|\mathcal{T}(\gamma, 2g')| = n_c(\lceil \gamma/2 \rceil, g')$$

Proof: We first find a lower bound on $|\mathcal{T}(\gamma, 2g')|$ and then exhibit a potential trapping set of size $n_c(\lceil \gamma/2 \rceil, g')$. We begin with the following lemma.

Lemma 3: $|\mathcal{T}(\gamma, 2g')| \geq n_c(\lceil \gamma/2 \rceil, g')$.

Proof: Let $\mathcal{T}_1$ be a trapping set with $|\mathcal{T}_1| < n_c(\lceil \gamma/2 \rceil, g')$ and let G_1 denote the induced subgraph of $\mathcal{T}_1$. We can construct a $(\lceil \gamma/2 \rceil, g'')$- cage graph $(g'' \geq g)$ with $|\mathcal{T}_1| < n_c(\lceil \gamma/2 \rceil, g')$ nodes by removing edges (if necessary) from the inverse edge-vertex of G_1 which is a contradiction. ∎
We now exhibit a potential trapping set of size $n_c(\lceil \gamma/2 \rceil, g')$. Let $G_{ev}(\lceil \gamma/2 \rceil, g')$ be the edge-vertex incidence graph of a $G(\lceil \gamma/2 \rceil, g')$. Note that $G_{ev}(\lceil \gamma/2 \rceil, g')$ is a left regular bipartite graph with $n_c(\lceil \gamma/2 \rceil, g')$ variable nodes of degree $\lceil \gamma/2 \rceil$ and all checks have degree two. Now consider $G_{ev,\gamma}(\lceil \gamma/2 \rceil, g')$, the γ augmented graph of $G_{ev}(\lceil \gamma/2 \rceil, g')$. It can be seen that $G_{ev,\gamma}(\lceil \gamma/2 \rceil, g')$ is a potential trapping set. ∎

Corollary 2: Let $\mathcal{C}$ be an LDPC code with column-weight γ and girth $2g'$. Then the bit flipping algorithm cannot be guaranteed to correct all error patterns of weight more than or equal to $n_c(\lceil \gamma/2 \rceil, g')$.

V. Discussion

We derived lower bounds and upper bounds on the guaranteed error correction capability of left regular LDPC codes. The lower bounds we derived in this paper are weak. However, extremal graphs avoiding three, four and five cycles have been studied in great detail (see [22], [23]) and these results can be used to derive tighter bounds when the girth is eight, ten or twelve. Also, since an expansion factor of $3\gamma/4$ is not necessary (see [7, Theorem 24]), it is possible that tighter lower bounds can be derived for some cases. The results can be extended to Gallager A and Gallager B algorithms as well. It should be noted that the necessary and sufficient conditions for a set to be trapping set for Gallager A/B algorithms are similar (depending on the message passing rules) to those in Theorem 4. Our approach can be used to derive bounds on the guaranteed erasure recovery capability for iterative decoding on the BEC by finding number of variable nodes which expand by a factor of $\gamma/2$. In [4], the bounds on the guaranteed erasure recovery capability were derived based on the size of smallest stopping set. Both approaches give the same bound, which also coincide with the bounds given by Tanner in [20] for the minimum distance.

References

[1] T. J. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[2] T. J. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[3] C. Di, D. Proietti, T. Richardson, E. Telatar, and R. Urbanke, "Finite length analysis of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570–1579, June 2002.

[4] A. Orlitsky, R. Urbanke, K. Viswanathan, and J. Zhang, "Stopping sets and the girth of tanner graphs," in *Proc. of IEEE International Symposium on Information Theory*, 2002, p. 2.

[5] R. G. Gallager, *Low Density Parity Check Codes*. Cambridge, MA: M.I.T. Press, 1963.

[6] V. V. Zyablov and M. S. Pinsker, "Estimation of the error-correction complexity for Gallager low-density codes," *Problems of Information Transmission*, vol. 11, no. 1, pp. 18–28, 1976.

[7] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.

[8] D. Burshtein and G. Miller, "Expander graph arguments for message-passing algorithms," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 782–790, Feb. 2001.

[9] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inform. Theory*, vol. 53, no. 1, pp. 82–89, Jan. 2007.

[10] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 954–972, March 2005.

[11] D. Burshtein, "On the error correction of regular LDPC codes using the flipping algorithm," in *Proc. of IEEE International Symposium on Information Theory*, June 2007, pp. 226–230.

[12] N. Alon, "Spectral techniques in graph algorithms," in *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics*. London, UK: Springer-Verlag, 1998, pp. 206–215.

[13] N. Biggs, *Algebraic graph theory*. Cambridge: Cambridge University Press, 1993.

[14] T. J. Richardson, "Error floors of LDPC codes," in *Proc. of 41st Annual Allerton Conf. on Communications, Control and Computing*, 2003, pp. 1426–1435.

[15] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," May 2007, accepted for IEEE Transactions on Information Theory. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0512078

[16] S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic, "Error floors of LDPC codes on the binary symmetric channel," in *Proc. of IEEE International Conference on Communications*, vol. 3, June 11-15 2006, pp. 1089–1094.

[17] E. W. Weisstein, "Cage graph." [Online]. Available: http://mathworld.wolfram.com/CageGraph.html

[18] B. Bollobas, *Extremal graph theory*. London: Academic Press Inc., 1978.

[19] A. Shokrollahi, "An introduction to low-density parity-check codes," in *Theoretical aspects of computer science: advanced lectures*. New York, NY, USA: Springer-Verlag New York, Inc., 2002, pp. 175–197.

[20] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981.

[21] N. Alon, S. Hoory, and M. Linial, "The moore bound for irregular graphs," *Graphs and Combinatorics*, vol. 18, no. 1, pp. 53–57, 2002.

[22] D. K. Garnick, Y. H. H. Kwong, and F. Lazebnik, "Extremal graphs without three-cycles or four-cycles," *J. Graph Theory*, vol. 17, no. 5, pp. 633–645, 1993.

[23] Y. Yuansheng, L. Xiaohui, D. Guocheng, and Z. Yongxiang, "Extremal graphs without three-cycles, four-cycles or five-cycles," *Utilitas Mathematica*, vol. 66, pp. 249–266, 2004.